Near-Eye Display Eye Tracking via Convolutional Neural Networks

Robert Konrad Shikhar Shrestha Paroma Varma

Introduction

Near-eye displays, such as virtual reality (VR) and augmented reality (AR) systems, are beginning to enter the consumer market. The overall goal of such displays is to create as immersive an experience as possible for the user, while being as comfortable as possible. The major achievement that Oculus and other competitors have capitalized on is the progression of cell-phone display technology. The cell-phone industry has encouraged LCD panel manufacturers to create very high-resolution, low-latency screens in small form-factors because of the enormous market. The VR and AR industries were directly able to tap into these advances in order to create basic wide field of view, high resolution products that do not incur motion sickness with the movement of the head, by combining the low-latency of the panels with some additional time-warping algorithms. Although initial prototypes create an immersive experience, and have begun selling, they only address the fundamental problems necessary for users to avoid nausea when using the devices. There is much more to be done in creating a more immersive experience.

One cue that VR and AR displays are currently missing is depth of field rendering. Depth of field is the amount of distance between the nearest and farthest objects that appear acceptably in focus when we focus to a certain distance. We experience this in our daily lives constantly. For example if I look to something that is very far away, objects closer to me will appear blurry because of the nature of our optical system. If I look to an object that is very close, for example holding my hand in front of me, objects that are far appear to be blurry. This depth of field blur is a strong cue used for depth perception, but also a technique increases immersion and a “fun” factor in virtual reality if it can be implemented in a gaze-contingent manner in real-time[1,2]. Rendering a depth of field into a scene in real-time is not difficult as there has been much work done in this area for video games previously [3,4,5,6]. The main challenge that we attempt to address with this project is performing gaze-tracking in a scene, particularly for near-eye display.

Although there has been much work done in gaze-tracking, most of it has been done on users looking at a screen from a distance. The camera was able to capture the full view of the pupil and iris, which is important for traditional gaze-tracking algorithms, and performed reasonably well [references here?]. The additional difficulty when implementing such a technique for near-eye displays is that there is simply much less room to work with, because many of these displays are head-mounted. A camera must be installed inside of the head-mounted display, without obstructing the view of the user while simultaneously being able to capture the full view of the pupil and iris regardless of the gaze of the user. This has proven to be a challenging issue, as there is only one company, SMI, that is able to implement near-eye display eye tracking, at the cost of $10,000. You are required to send your Oculus DK2 to the company, which they will take apart to install a camera in each eye. We seek a more elegant solution to the problem, and one that might be able to place a camera in a sub-optimal position where it might not capture a full view of the eye's pupil and iris. We will not use traditional gaze-tracking algorithms, but instead use a series of input/output pairs to train a neural network that will predict where a user is looking based on partial images of the user's pupil and iris.

Our Proposal

A CNN based approach to eye tracking instead of feature based methods that are pervasive in the industry could be very powerful as it has the ability to account for variability in the images. The main challenge with eye tracking systems is that people have different skin tone, eye color, size and proportions and the FoV isn't always perfectly aligned. This creates problems with feature based tracking methods as the detectors are often not able to detect feature points of the user's eye.

The core hypothesis of this work is to use a CNN that has been trained on a large near eye display eye tracking dataset that we will capture ourselves and then use a brief training exercise to fine-tune the model for each user upon wearing the headset to calibrate and improve tracking accuracy.

We propose to capture a large dataset by performing a user study where they follow a moving marker on a screen for 5 minutes. As the location of the marker is known, this results in 30X60X5 frames per user. The captured dataset will then be augmented using standard data augmentation methods and split into training, test and cross-validation sets.

A small CNN such as the LeNet will then be trained to the dataset and we can iterate the model structure and hyperparameters using the cross-validation data.

Caffe/TensorFlow would be used for training the model and inference which will then be ported to a small linux SBC to tracks the user's gaze in real-time while he/she is wearing the near eye display.

Milestones

The Project can be divided into two major tasks:

- Dataset collection and preparation (4/20-5/10)
    - Writing the capture script
    - Conduct user study
    - Pre-process data (channel normalization etc)
    - Augment dataset with noise, transformations etc
    - Convert into appropriate data format (lmdb)

- CNN training and validation (5/10-6/1)
    - Define model architecture
    - Train and cross-validate
    - Fine-tune
    - Test and iterate

References

[1] R. Mantiuk, B. Bazyluk, and A. Tomaszewska. 2011. Gaze-Dependent depth-of-field effect rendering in virtual environments. In *Proceedings of the Second international conference on Serious Games Development and Applications* (SGDA'11). Springer-Verlag, Berlin, Heidelberg, 1-12.

[2] M. Mauderer, S. Conte, M. a. Nacenta, and D. Vishwanath, “Depth perception with gaze-contingent depth of field,” Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI ’14, pp. 217–226, 2014.

[3] B. a. Barsky and T. J. Kosloff, “Algorithms for Rendering Depth of Field Effects in Computer Graphics,” World Scientific and Engineering Academy and Society (WSEAS), pp. 999–1010, 2008.

[4] T. Zhou, J. X. Chen, and M. Pullen, “Accurate Depth of Field Simulation in Real Time,” Time, vol. 26, no. 1, pp. 15–23, 2007.

[5] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in Computer Vision, 1998. Sixth International Conference on, pp. 839–846, Jan 1998.

[6] S. Xu, X. Mei, W. Dong, X. Sun, X. Shen, and X. Zhang, “Depth of Field Rendering via Adaptive Recursive Filtering,” 2014.